Large Margin Multiclass Gaussian Classification with Differential Privacy

Manas A. Pathak and Bhiksha Raj

Carnegie Mellon University
{manasp, bhiksha}@cs.cmu.edu

Abstract. As increasing amounts of sensitive personal information is aggregated into data repositories, it has become important to develop mechanisms for processing the data without revealing information about individual data instances. The differential privacy model provides a framework for the development and theoretical analysis of such mechanisms. In this paper, we propose an algorithm for learning a discriminatively trained multi-class Gaussian classifier that satisfies differential privacy using a large margin loss function with a perturbed regularization term. We present a theoretical upper bound on the excess risk of the classifier introduced by the perturbation.

1 Introduction

In recent years, vast amounts of personal data is being aggregated in the form of medical, financial records, social networks, and government census data. As these often contain sensitive information, a database curator interested in releasing a function such as a statistic evaluated over the data is faced with the prospect that it may lead to a breach of privacy of the individuals who contributed to the database. It is therefore important to develop techniques for retrieving desired information from a dataset without revealing any information about individual data instances. *Differential privacy* [1] is a theoretical model proposed to address this issue. A query mechanism evaluated over a dataset is said to satisfy differential privacy if it is likely to produce the same output on a dataset differing by at most one element. This implies that an adversary having complete knowledge of all data instances but one along with *a priori* information about the remaining instance, is not likely to be able to infer any more information about the remaining instance by observing the output of the mechanism.

One of the most common applications for such large data sets such as the ones mentioned above is for training classifiers that can be used to categorize new data. If the training data contains private data instances, an adversary should not be able to learn anything about the individual training dataset instances by analyzing the output of the classifier. Recently, mechanisms for learning differentially private classifiers have been proposed for logistic regression [2]. In this method, the objective function which is minimized by the classification algorithm is modified by adding a linear perturbation term. Compared to the original classifier, there is an additional error introduced by the perturbation term in the

differentially private classifier. It is important to have an upper bound on this error as a cost of preserving privacy.

The work mentioned above is largely restricted to binary classification, while multi-class classifiers are more useful in many practical situations. In this paper, we propose an algorithm for learning multi-class Gaussian classifiers which satisfies differential privacy. Gaussian classifiers that model the distributions of individual classes as being generated from Gaussian distribution or a mixture of Gaussian distributions [3] are commonly used as multi-class classifiers. We use a large margin discriminative algorithm for training the classifier introduced by Sha and Saul [4]. To ensure that the learned multi-class classifier preserves differential privacy, we modify the objective function by introducing a perturbed regularization term.

2 Differential Privacy

In recent years, the differential privacy model proposed by Dwork, *et al.* [1] has emerged as a robust standard for data privacy. It originated from the statistical database model, where the dataset D is a collection of elements and a randomized *query mechanism* M produces a response when performed on a given dataset. Two datasets D and D' differing by at most one element are said to be *adjacent*. There are two proposed definitions for adjacent datasets one based on symmetric difference – D' containing of one entry less than D, and one based on substitution – one entry of D' differs in value from D. We use the substitution definition of adjacency previously used by [5,2], where the one entry of the dataset $D = \{x_1, \ldots, x_{n-1}, x_n\}$ is modified to result in an adjacent dataset $D' = \{x_1, \ldots, x_{n-1}, x'_n\}$. The query mechanism M is said to satisfy differential privacy if the probability of M resulting in a solution S when performed on a dataset D is very close to the probability of M resulting in the same solution S when executed on an adjacent dataset D'. Assuming the query mechanism to be a function $M : D \mapsto \text{range}(M)$ with a probability function P defined over the space of M, differential privacy is formally defined as follows.

Definition 1. *A randomized function M satisfies ϵ-differential privacy if for all adjacent datasets D and D' and for any $S \in range(M)$,*

$$\left| \log \frac{P\left(M(D) = S\right)}{P\left(M(D') = S\right)} \right| \leq \epsilon.$$

The value of the ϵ parameter, which is referred to as *leakage*, determines the degree of privacy. As there is always a trade-off between privacy and utility, the choice of ϵ is motivated by the requirements of the application.

In a machine learning setting, the query mechanism can be thought of as an algorithm learning the classification, regression or density estimation rule which is evaluated over the training dataset. The output of an algorithm satisfying differential privacy is likely to be same when the value of any single dataset instance is modified, and therefore, no additional information can be obtained about any

individual training data instances with certainty by observing the output of the learning algorithm, beyond what is already known to an adversary. Differential privacy is a strong definition of privacy – it provides *ad omnia* guarantee as opposed to most other models that provide *ad hoc* guarantees against specific set of attacks and adversarial behaviors.

2.1 Related Work

The earlier work on differential privacy was related to functional approximations for simple data mining tasks and data release mechanisms [6,7,8,9]. Although many of these works have connection to machine learning problems, more recently the design and analysis of machine learning algorithms satisfying differential privacy has been actively studied. Kasiviswanathan, *et al.* [5] present a framework for converting a general agnostic PAC learning algorithm to an algorithm that satisfies privacy constraints. Chaudhuri and Monteleoni [2] use the exponential mechanism [10] to create a differentially private logistic regression classifier by adding Laplace noise to the estimated parameters. They propose another differentially private formulation which involves modifying the objective function of the logistic regression classifier by adding a linear term scaled by Laplace noise. The second formulation is advantageous because it is independent of the classifier sensitivity which difficult to compute in general and it can be shown that using a perturbed objective function introduces a lower error as compared to the exponential mechanism.

However, the above mentioned differentially private classification algorithms only address the problem of binary classification. Although it is possible to extend binary classification algorithms to multi-class using techniques like one-vs-all, it is much more expensive to do so as compared to a naturally multi-class classification algorithm. Jagannathan, *et al.* [11] present a differentially private random decision tree learning algorithm which can be applied to multi-class classification. Their approach involves perturbing leaf nodes using the sensitivity method, and they do not provide theoretical analysis of excess risk of the perturbed classifier. In this paper, we propose a modification to the naturally multi-class large margin Gaussian classification algorithm [4,12].

3 Large Margin Gaussian Classifiers

We investigate the large margin multi-class classification algorithm introduced by Sha and Saul [4]. The training dataset $(\boldsymbol{x}, \boldsymbol{y})$[1] contains n iid d-dimensional training data instances $\boldsymbol{x}_i \in \mathbb{R}^d$ each with labels $y_i \in \{1, \ldots, C\}$. We consider the setting where each class is modeled as a single Gaussian ellipsoid. Each class ellipsoid is parametrized by the centroid $\boldsymbol{\mu}_c \in \mathbb{R}^d$, the inverse covariance matrix $\boldsymbol{\Psi}_c \in \mathbb{R}^{d \times d}$, and a scalar offset $\theta_c \geq 0$. The decision rule is to assign an instance

[1] Notation: vectors and matrices are denoted by **boldface**.

$\boldsymbol{x}_i$ to the class having smallest Mahalanobis distance [13] with the scalar offset from $\boldsymbol{x}_i$ to the centroid of that class.

$$y_i = \underset{c}{\operatorname{argmin}} \, (\boldsymbol{x}_i - \boldsymbol{\mu}_c)^T \boldsymbol{\Psi}_c (\boldsymbol{x}_i - \boldsymbol{\mu}_c) + \theta_c. \tag{1}$$

To simplify the notation, we expand $(\boldsymbol{x}_i - \boldsymbol{\mu}_c)^T \boldsymbol{\Psi}_c (\boldsymbol{x}_i - \boldsymbol{\mu}_c)$ and collect the parameters for each class as the following $(d+1) \times (d+1)$ *positive semidefinite* matrix

$$\boldsymbol{\Phi}_c = \begin{bmatrix} \boldsymbol{\Psi}_c & -\boldsymbol{\Psi}_c \boldsymbol{\mu}_c \\ -\boldsymbol{\mu}_c^T \boldsymbol{\Psi}_c & \boldsymbol{\mu}_c^T \boldsymbol{\Psi}_c \boldsymbol{\mu}_c + \theta_c \end{bmatrix} \tag{2}$$

and also append a unit element to each d-dimensional vector $\boldsymbol{x}_i$. The decision rule for a data instance $\boldsymbol{x}_i$ simplifies to

$$y_i = \underset{c}{\operatorname{argmin}} \, \boldsymbol{x}_i^T \boldsymbol{\Phi}_c \boldsymbol{x}_i. \tag{3}$$

The discriminative training procedure involves estimating a set of positive semidefinite matrices $\{\boldsymbol{\Phi}_1, \ldots, \boldsymbol{\Phi}_C\}$ from the training data $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$ which optimize the performance on the decision rule mentioned above. We apply the large margin intuition that the optimal classifier must maximize the distance of training data instances from the decision boundaries. This leads to the classification algorithm being robust to outliers with provably strong generalization guarantees. Formally, we require that for each training data instance $\boldsymbol{x}_i$ with label y_i, the distance from $\boldsymbol{x}_i$ to the centroid of class y_i is at least less than its distance from centroids of all other classes by one.

$$\forall c \neq y_i : \boldsymbol{x}_i^T \boldsymbol{\Phi}_c \boldsymbol{x}_i \;\geq\; 1 + \boldsymbol{x}_i^T \boldsymbol{\Phi}_{y_i} \boldsymbol{x}_i.$$

Analogous to support vector machines, the training algorithm is an optimization problem minimizing the *hinge loss* denoted by $[f]_+ = \max(0, f)$, with a linear penalty for incorrect classification. We use the sum of traces of inverse covariance matrices for each classes as a *regularization* term. The regularization requires that if we can learn a classifier which labels every training data instance correctly, we choose the one with the lowest inverse covariance or highest covariance for each class ellipsoid as this prevents the classifier from over-fitting. The parameter λ controls the trade off between the loss function and the regularization.

$$J(\boldsymbol{\Phi}, \boldsymbol{x}, \boldsymbol{y}) = \sum_i \sum_{c \neq y_i} \left[1 + \boldsymbol{x}_i^T (\boldsymbol{\Phi}_{y_i} - \boldsymbol{\Phi}_c) \boldsymbol{x}_i\right]_+ + \lambda \sum_c \operatorname{trace}(\boldsymbol{\Psi}_c). \tag{4}$$

The inverse covariance matrix $\boldsymbol{\Psi}_c$ is contained in the upper left size $d \times d$ block of the matrix $\boldsymbol{\Phi}_c$. We replace it with $\mathbf{I}_{\boldsymbol{\Phi}} \boldsymbol{\Phi}_c \mathbf{I}_{\boldsymbol{\Phi}}$, where $\mathbf{I}_{\boldsymbol{\Phi}}$ is the truncated size $(d+1) \times (d+1)$ identity matrix with the last diagonal element $I_{\Phi_{d+1,d+1}}$ set to zero. The optimization problem becomes

$$\begin{aligned} J(\boldsymbol{\Phi}, \boldsymbol{x}, \boldsymbol{y}) &= \sum_i \sum_{c \neq y_i} \left[1 + \boldsymbol{x}_i^T (\boldsymbol{\Phi}_{y_i} - \boldsymbol{\Phi}_c) \boldsymbol{x}_i\right]_+ + \lambda \sum_c \operatorname{trace}(\mathbf{I}_{\boldsymbol{\Phi}} \boldsymbol{\Phi}_c \mathbf{I}_{\boldsymbol{\Phi}}) \\ &= L(\boldsymbol{\Phi}, \boldsymbol{x}, \boldsymbol{y}) + N(\boldsymbol{\Phi}). \end{aligned} \tag{5}$$

The hinge loss being non-differentiable is not very convenient for our analysis; we replace it with a surrogate loss function called Huber loss l_h [14] which has similar characteristics to the hinge loss for small values of h.

$$\ell_h(\mathbf{\Phi}_c, x_i, y_i) = \begin{cases} 0 & \text{if } \boldsymbol{x}_i^T(\mathbf{\Phi}_c - \mathbf{\Phi}_{y_i})\boldsymbol{x}_i > h, \\ \frac{1}{4h}\left[h - \boldsymbol{x}_i^T(\mathbf{\Phi}_{y_i} - \mathbf{\Phi}_c)\boldsymbol{x}_i\right]^2 & \text{if } |\boldsymbol{x}_i^T(\mathbf{\Phi}_c - \mathbf{\Phi}_c)\boldsymbol{x}_i| \leq h \\ -\boldsymbol{x}_i^T(\mathbf{\Phi}_{y_i} - \mathbf{\Phi}_c)\boldsymbol{x}_i & \text{if } \boldsymbol{x}_i^T(\mathbf{\Phi}_c - \mathbf{\Phi}_{y_i})\boldsymbol{x}_i < -h. \end{cases} \quad (6)$$

The objective function is convex function of positive semidefinite matrices $\mathbf{\Phi}_c$. The optimization can be formulated as a semidefinite programming problem [15] and be solved efficiently using interior point methods.

The large margin classification framework can be easily extended to modeling each class with a mixture of Gaussians. Similar to support vector machines, when training with non-separable data, we can introduce slack parameters to permit margin violations. These extensions do not change the basic characteristics of the learning algorithm. The optimization problem remains to be a convex semidefinite program with piecewise linear terms and is equally tractable. For simplicity, we restrict our discussion to single Gaussians and hard margins in this paper. As we shall see, it is easy to extend our proposed modifications to these cases.

4 Differentially Private Large Margin Gaussian Classifiers

We modify the large margin Gaussian classification formulation to satisfy differential privacy by introducing a perturbation term in the objective function. As we will see in Section 5.2, this modification leads to a classifier that preserves differential privacy.

We generate the size $(d+1) \times (d+1)$ perturbation matrix $\mathbf{b}$ with density

$$P(\mathbf{b}) \propto \exp\left(-\frac{\epsilon}{2}\|\mathbf{b}\|\right), \quad (7)$$

where $\|\cdot\|$ is the Frobenius norm (element-wise ℓ_2 norm) and ϵ is the privacy parameter. One method of generating such a $\mathbf{b}$ matrix is to sample the norm $\|\mathbf{b}\|$ from $\mathit{\Gamma}\left((d+1)^2, \frac{2}{\epsilon}\right)$ and the direction of $\mathbf{b}$ at random.

Our proposed learning algorithm minimizes the following objective function $J_p(\mathbf{\Phi}, \boldsymbol{x}, \boldsymbol{y})$, where the subscript p denotes privacy.

$$\begin{aligned} J_p(\mathbf{\Phi}, \boldsymbol{x}, \boldsymbol{y}) &= L(\mathbf{\Phi}, \boldsymbol{x}, \boldsymbol{y}) + \lambda \sum_c \operatorname{trace}(\mathbf{I}_{\mathbf{\Phi}} \mathbf{\Phi}_c \mathbf{I}_{\mathbf{\Phi}}) + \sum_c \sum_{ij} b_{ij} \mathit{\Phi}_{cij} \\ &= J(\mathbf{\Phi}, \boldsymbol{x}, \boldsymbol{y}) + \sum_c \sum_{ij} b_{ij} \mathit{\Phi}_{cij}. \end{aligned} \quad (8)$$

As the dimensionality of the perturbation matrix $\mathbf{b}$ is same as that of the classifier parameters $\mathbf{\Phi}_c$, the parameter space of $\mathbf{\Phi}$ does not change after perturbation. In other words, given two datasets $(\boldsymbol{x}, \boldsymbol{y})$ and $(\boldsymbol{x}', \boldsymbol{y}')$, if $\mathbf{\Phi}^{\mathbf{P}}$ minimizes

$J_p(\boldsymbol{\Phi}, \boldsymbol{x}, \boldsymbol{y})$, it is always possible to have $\boldsymbol{\Phi^p}$ minimize $J_p(\boldsymbol{\Phi}, \boldsymbol{x'}, \boldsymbol{y'})$. This is a necessary condition for the classifier $\boldsymbol{\Phi^p}$ satisfying differential privacy.

Furthermore, as the perturbation term is convex and positive semidefinite, the perturbed objective function $J_p(\boldsymbol{\Phi}, \boldsymbol{x}, \boldsymbol{y})$ has the same properties as the unperturbed objective function $J(\boldsymbol{\Phi}, \boldsymbol{x}, \boldsymbol{y})$. Also, the perturbation does not introduce any additional computational cost as compared to the original algorithm.

5 Theoretical Analysis

5.1 Proof of Differential Privacy

In the following theorem, we prove that the classifier minimizing the perturbed optimization function $J_p(\boldsymbol{\Phi}, \boldsymbol{x}, \boldsymbol{y})$ satisfies ϵ-differential privacy. Given the dataset $(\boldsymbol{x}, \boldsymbol{y}) = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_{n-1}, y_{n-1}), (\boldsymbol{x}_n, y_n)\}$, the probability of learning the classifier $\boldsymbol{\Phi}^p$ is close to the the probability of learning the same classifier $\boldsymbol{\Phi}^p$ given its adjacent dataset $(\boldsymbol{x'}, \boldsymbol{y'}) = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_{n-1}, y_{n-1}), (\boldsymbol{x'}_n, y'_n)\}$ differing wlog on the n^{th} instance. As we mentioned in the previous section, it is always possible to find such a classifier $\boldsymbol{\Phi}^p$ minimizing both $J_p(\boldsymbol{\Phi}, \boldsymbol{x}, \boldsymbol{y})$ and $J_p(\boldsymbol{\Phi}, \boldsymbol{x'}, \boldsymbol{y'})$ due to the perturbation matrix being in the same space as the optimization parameters.

Our proof requires a strictly convex perturbed objective function resulting in a unique solution $\boldsymbol{\Phi}^p$ minimizing it. This in turn requires that the loss function $L(\boldsymbol{\Phi}, \boldsymbol{x}, y)$ is strictly convex and differentiable, and the regularization term $N(\boldsymbol{\Phi})$ is convex. These seemingly strong constraints are satisfied by many commonly used classification algorithms such as logistic regression, support vector machines, and our general perturbation technique can be extended to those algorithms. In our proposed algorithm, the Huber loss is by definition a differentiable function and the trace regularization term is convex and differentiable. Additionally, we require that the difference in the gradients of $L(\boldsymbol{\Phi}, \boldsymbol{x}, y)$ calculated over for two adjacent training datasets is bounded. We prove this property in Lemma 1 given in the appendix.

Theorem 1. *For any two adjacent training datasets $(\boldsymbol{x}, \boldsymbol{y})$ and $(\boldsymbol{x'}, \boldsymbol{y'})$, the classifier $\boldsymbol{\Phi}^p$ minimizing the perturbed objective function $J_p(\boldsymbol{\Phi}, \boldsymbol{x}, \boldsymbol{y})$ satisfies differential privacy.*

$$\left| \log \frac{P(\boldsymbol{\Phi}^p | \boldsymbol{x}, \boldsymbol{y})}{P(\boldsymbol{\Phi}^p | \boldsymbol{x'}, \boldsymbol{y'})} \right| \leq \epsilon',$$

where $\epsilon' = \epsilon + k$ for a constant factor $k = \log\left(1 + \frac{2\alpha}{n\lambda} + \frac{\alpha^2}{n^2\lambda^2}\right)$ with a constant value of α.

Proof. As $J(\boldsymbol{\Phi}, \boldsymbol{x}, \boldsymbol{y})$ is convex and differentiable, there is a unique solution $\boldsymbol{\Phi}^*$ that minimizes it. As the perturbation term $\sum_c \sum_{ij} b_{ij} \Phi_{cij}$ is also convex and differentiable, the perturbed objective function $J_p(\boldsymbol{\Phi}, \boldsymbol{x}, \boldsymbol{y})$ also has a unique

solution $\mathbf{\Phi}^p$ that minimizes it. Differentiating $J_p(\mathbf{\Phi}, \boldsymbol{x}, \boldsymbol{y})$ wrt $\mathbf{\Phi}_c$, we have

$$\frac{\partial}{\partial \mathbf{\Phi}_c} J_p(\mathbf{\Phi}, \boldsymbol{x}, \boldsymbol{y}) = \frac{\partial}{\partial \mathbf{\Phi}_c} L(\mathbf{\Phi}, \boldsymbol{x}, \boldsymbol{y}) + \lambda \mathbf{I}_{\mathbf{\Phi}} + \mathbf{b}. \tag{9}$$

Substituting the optimal $\mathbf{\Phi}_c^p$ in the derivative gives us

$$\lambda \mathbf{I}_{\mathbf{\Phi}} + \mathbf{b} = -\frac{\partial}{\partial \mathbf{\Phi}_c} L(\mathbf{\Phi}^p, \boldsymbol{x}, \boldsymbol{y}).$$

This relation shows that two different values of $\mathbf{b}$ cannot result in the same optimal $\mathbf{\Phi}^p$. As the perturbed objective function $J_p(\mathbf{\Phi}, \boldsymbol{x}, \boldsymbol{y})$ is also convex and differentiable, there is a bijective map between the perturbation $\mathbf{b}$ and the unique $\mathbf{\Phi}^p$ minimizing $J_p(\mathbf{\Phi}, \boldsymbol{x}, \boldsymbol{y})$.

Let $\mathbf{b_1}$ and $\mathbf{b_2}$ be the two perturbations applied when training with the adjacent datasets $(\boldsymbol{x}, \boldsymbol{y})$ and $(\boldsymbol{x}', \boldsymbol{y}')$, respectively. Assuming that we obtain the same optimal solution $\mathbf{\Phi}^p$ while minimizing both $J_p(\mathbf{\Phi}, \boldsymbol{x}, \boldsymbol{y})$ with perturbation $\mathbf{b}_1$ and $J_p(\mathbf{\Phi}, \boldsymbol{x}, \boldsymbol{y})$ with perturbation $\mathbf{b}_2$,

$$\begin{aligned}
\lambda \mathbf{I}_{\mathbf{\Phi}} + \mathbf{b}_1 &= -\frac{\partial}{\partial \mathbf{\Phi}_c} L(\mathbf{\Phi}^p, \boldsymbol{x}, \boldsymbol{y}), \\
\lambda \mathbf{I}_{\mathbf{\Phi}} + \mathbf{b}_2 &= -\frac{\partial}{\partial \mathbf{\Phi}_c} L(\mathbf{\Phi}^p, \boldsymbol{x}', \boldsymbol{y}'), \\
\mathbf{b}_1 - \mathbf{b}_2 &= \frac{\partial}{\partial \mathbf{\Phi}_c} L(\mathbf{\Phi}^p, \boldsymbol{x}', \boldsymbol{y}') - \frac{\partial}{\partial \mathbf{\Phi}_c} L(\mathbf{\Phi}^p, \boldsymbol{x}, \boldsymbol{y}).
\end{aligned} \tag{10}$$

We apply Lemma 1 after taking Frobenius norm on both sides.

$$\begin{aligned}
\|\mathbf{b}_1 - \mathbf{b}_2\| &= \left\| \frac{\partial}{\partial \mathbf{\Phi}_c} L(\mathbf{\Phi}^p, \boldsymbol{x}', \boldsymbol{y}') - \frac{\partial}{\partial \mathbf{\Phi}_c} L(\mathbf{\Phi}^p, \boldsymbol{x}, \boldsymbol{y}) \right\| \\
&= \left\| \sum_{i=1}^{n-1} \frac{\partial}{\partial \mathbf{\Phi}_c} L(\mathbf{\Phi}^p, \boldsymbol{x}_i, y_i) + \frac{\partial}{\partial \mathbf{\Phi}_c} L(\mathbf{\Phi}^p, \boldsymbol{x}'_n, y'_n) \right. \\
&\qquad \left. - \sum_{i=1}^{n-1} \frac{\partial}{\partial \mathbf{\Phi}_c} L(\mathbf{\Phi}^p, \boldsymbol{x}_i, y_i) - \frac{\partial}{\partial \mathbf{\Phi}_c} L(\mathbf{\Phi}^p, \boldsymbol{x}_n, y_n) \right\| \\
&= \left\| \frac{\partial}{\partial \mathbf{\Phi}_c} L(\mathbf{\Phi}^p, \boldsymbol{x}'_n, y'_n) - \frac{\partial}{\partial \mathbf{\Phi}_c} L(\mathbf{\Phi}^p, \boldsymbol{x}_n, y_n) \right\| \leq 2.
\end{aligned}$$

Using this property, we can calculate the ratio of densities of drawing the perturbation matrices $\mathbf{b}_1$ and $\mathbf{b}_2$ as

$$\frac{P(\mathbf{b} = \mathbf{b}_1)}{P(\mathbf{b} = \mathbf{b}_2)} = \frac{\frac{1}{\text{surf}(\|\mathbf{b}_1\|)} \|\mathbf{b}_1\|^d \exp\left[-\frac{\epsilon}{2}\|\mathbf{b}_1\|\right]}{\frac{1}{\text{surf}(\|\mathbf{b}_2\|)} \|\mathbf{b}_2\|^d \exp\left[-\frac{\epsilon}{2}\|\mathbf{b}_2\|\right]},$$

where $\text{surf}(\|\mathbf{b}\|)$ is the surface area of the $(d+1)$-dimensional hypersphere with radius $\|\mathbf{b}\|$. As $\text{surf}(\|\mathbf{b}\|) = \text{surf}(1)\|\mathbf{b}\|^d$, where $\text{surf}(1)$ is the area of the unit

$(d+1)$-dimensional hypersphere, the ratio of the densities becomes

$$\frac{P(\mathbf{b}=\mathbf{b}_1)}{P(\mathbf{b}=\mathbf{b}_2)} = \exp\left[\frac{\epsilon}{2}(\|\mathbf{b}_2\| - \|\mathbf{b}_1\|)\right] \leq \exp\left[\frac{\epsilon}{2}\|\mathbf{b}_2 - \mathbf{b}_1\|\right] \leq \exp(\epsilon). \tag{11}$$

The ratio of the densities of learning $\mathbf{\Phi}^p$ using the adjacent datasets $(\boldsymbol{x}, \boldsymbol{y})$ and $(\boldsymbol{x}', \boldsymbol{y}')$ is given by

$$\frac{P(\mathbf{\Phi}^p|\boldsymbol{x}, \boldsymbol{y})}{P(\mathbf{\Phi}^p|\boldsymbol{x}', \boldsymbol{y}')} = \frac{P(\mathbf{b}=\mathbf{b}_1)}{P(\mathbf{b}=\mathbf{b}_2)} \frac{|\det(\mathbf{J}(\mathbf{\Phi}^p \rightarrow \mathbf{b}_1|\boldsymbol{x}, \boldsymbol{y}))|^{-1}}{|\det(\mathbf{J}(\mathbf{\Phi}^p \rightarrow \mathbf{b}_2|\boldsymbol{x}', \boldsymbol{y}'))|^{-1}}, \tag{12}$$

where $\mathbf{J}(\mathbf{\Phi}^p \rightarrow \mathbf{b}_1|\boldsymbol{x}, \boldsymbol{y})$ and $\mathbf{J}(\mathbf{\Phi}^p \rightarrow \mathbf{b}_2|\boldsymbol{x}', \boldsymbol{y}')$ are the Jacobian matrices of the bijective mappings from $\mathbf{\Phi}^p$ to $\mathbf{b}_1$ and $\mathbf{b}_2$, respectively. Following a procedure identical to Theorem 2 of [16] (omitted due to lack of space), it can be shown that the ratio of Jacobian determinants is upper bounded by a constant factor $\exp(k) = 1 + \frac{2\alpha}{n\lambda} + \frac{\alpha^2}{n^2\lambda^2}$ for a constant value of α. Therefore, the ratio of the densities of learning $\mathbf{\Phi}^p$ using the adjacent datasets becomes

$$\frac{P(\mathbf{\Phi}^p|\boldsymbol{x}, \boldsymbol{y})}{P(\mathbf{\Phi}^p|\boldsymbol{x}', \boldsymbol{y}')} \leq \exp(\epsilon + k) = \exp(\epsilon'). \tag{13}$$

Similarly, we can show that the probability ratio is lower bounded by $\exp(-\epsilon')$, which together with Equation (13) satisfies the definition of differential privacy. □

5.2 Analysis of Excess Error

In the remainder of this section, we denote the terms $J(\mathbf{\Phi}, \mathbf{x}, \mathbf{y})$ and $L(\mathbf{\Phi}, \mathbf{x}, \mathbf{y})$ by $J(\mathbf{\Phi})$ and $L(\mathbf{\Phi})$ respectively for conciseness. To establish a bound on excess risk of the classifier given by the proposed algorithm minimizing the perturbed objective function, in Lemma 2 we show that the objective function $J(\mathbf{\Phi})$ satisfies strong convexity. The objective function $J(\mathbf{\Phi})$ contains the loss function $L(\mathbf{\Phi})$ computed over the training data $(\mathbf{x}, \mathbf{y})$ and the regularization term $N(\mathbf{\Phi})$ – this is known as the regularized *empirical risk* of the classifier $\mathbf{\Phi}$. In the following theorem, we establish a bound on the regularized empirical excess risk of the differentially private classifier minimizing the perturbed objective function over the classifier minimizing the unperturbed objective function.

Theorem 2. *With probability at least* $1-\delta$*, the regularized empirical excess risk of the classifier* $\mathbf{\Phi}^p$ *minimizing the perturbed objective function* $J_p(\mathbf{\Phi})$ *over the classifier* $\mathbf{\Phi}^*$ *minimizing the unperturbed objective function* $J(\mathbf{\Phi})$ *is bounded as*

$$J(\mathbf{\Phi}^p) \leq J(\mathbf{\Phi}^*) + \frac{8(d+1)^4 C}{\epsilon^2 \lambda} \log^2\left(\frac{d}{\delta}\right).$$

Proof. We use the definition of $J_p(\mathbf{\Phi}) = J(\mathbf{\Phi}) + \sum_c \sum_{ij} b_{ij} \Phi_{cij}$ and the optimality of $\mathbf{\Phi}^p$, *i.e.*, $J_p(\mathbf{\Phi}^p) \leq J_p(\mathbf{\Phi}^*)$.

$$\begin{aligned} J(\mathbf{\Phi}^p) + \sum_c \sum_{ij} b_{ij} \Phi^p_{cij} &\leq J(\mathbf{\Phi}^*) + \sum_c \sum_{ij} b_{ij} \Phi^*_{cij}, \\ J(\mathbf{\Phi}^p) &\leq J(\mathbf{\Phi}^*) + \sum_c \sum_{ij} b_{ij} (\Phi^*_{cij} - \Phi^p_{cij}). \end{aligned} \tag{14}$$

Using the strong convexity of $J(\mathbf{\Phi})$ as given by Lemma 2 and the optimality of $J(\mathbf{\Phi}^*)$, we have

$$\begin{aligned} J(\mathbf{\Phi}^*) \leq J\left(\frac{\mathbf{\Phi}^p + \mathbf{\Phi}^*}{2}\right) &\leq \frac{J(\mathbf{\Phi}^p) + J(\mathbf{\Phi}^*)}{2} - \frac{\lambda}{8} \sum_c \|\mathbf{\Phi}^*_c - \mathbf{\Phi}^p_c\|^2, \\ J(\mathbf{\Phi}^p) - J(\mathbf{\Phi}^*) &\geq \frac{\lambda}{4} \sum_c \|\mathbf{\Phi}^*_c - \mathbf{\Phi}^p_c\|^2. \end{aligned} \tag{15}$$

Similarly, using the strong convexity of $J_p(\mathbf{\Phi})$ and the optimality of $J_p(\mathbf{\Phi}^p)$,

$$\begin{aligned} J_p(\mathbf{\Phi}^p) \leq J_p\left(\frac{\mathbf{\Phi}^p + \mathbf{\Phi}^*}{2}\right) &\leq \frac{J_p(\mathbf{\Phi}^p) + J_p(\mathbf{\Phi}^*)}{2} - \frac{\lambda}{8} \sum_c \|\mathbf{\Phi}^p_c - \mathbf{\Phi}^*_c\|^2, \\ J_p(\mathbf{\Phi}^*) - J_p(\mathbf{\Phi}^p) &\geq \frac{\lambda}{4} \sum_c \|\mathbf{\Phi}^p_c - \mathbf{\Phi}^*_c\|^2. \end{aligned}$$

Substituting the definition $J_p(\mathbf{\Phi}) = J(\mathbf{\Phi}) + \sum_c \sum_{ij} b_{ij} \Phi_{cij}$,

$$\begin{aligned} J(\mathbf{\Phi}^*) + \sum_c \sum_{ij} b_{ij} \Phi^*_{cij} - J(\mathbf{\Phi}^p) - \sum_c \sum_{ij} b_{ij} \Phi^p_{cij} &\geq \frac{\lambda}{4} \sum_c \|\mathbf{\Phi}^*_c - \mathbf{\Phi}^p_c\|^2 \\ \sum_c \sum_{ij} b_{ij} (\Phi^*_{cij} - \Phi^p_{cij}) - (J(\mathbf{\Phi}^p) - J(\mathbf{\Phi}^*)) &\geq \frac{\lambda}{4} \sum_c \|\mathbf{\Phi}^*_c - \mathbf{\Phi}^p_c\|^2. \end{aligned}$$

Substituting the lower bound on $J(\mathbf{\Phi}^p) - J(\mathbf{\Phi}^*)$ given by Equation (15),

$$\begin{aligned} \sum_c \sum_{ij} b_{ij} (\Phi^*_{cij} - \Phi^p_{cij}) &\geq \frac{\lambda}{2} \sum_c \|\mathbf{\Phi}^*_c - \mathbf{\Phi}^p_c\|^2, \\ \left[\sum_c \sum_{ij} b_{ij} (\Phi^*_{cij} - \Phi^p_{cij})\right]^2 &\geq \frac{\lambda^2}{4} \left[\sum_c \|\mathbf{\Phi}^*_c - \mathbf{\Phi}^p_c\|^2\right]^2. \end{aligned} \tag{16}$$

Using the Cauchy-Schwarz inequality, we have,

$$\left[\sum_c \sum_{ij} b_{ij} (\Phi^*_{cij} - \Phi^p_{cij})\right]^2 \leq C \|\mathbf{b}\|^2 \sum_c \|\mathbf{\Phi}^*_c - \mathbf{\Phi}^p_c\|^2 \tag{17}$$

Combining this with Equation (16) gives us

$$C\|\mathbf{b}\|^2 \sum_c \|\mathbf{\Phi}_c^* - \mathbf{\Phi}_c^p\|^2 \geq \frac{\lambda^2}{4}\left[\sum_c \|\mathbf{\Phi}_c^* - \mathbf{\Phi}_c^p\|^2\right]^2,$$
$$\sum_c \|\mathbf{\Phi}_c^* - \mathbf{\Phi}_c^p\|^2 \leq \frac{4C}{\lambda^2}\|\mathbf{b}\|^2. \tag{18}$$

Combining this with Equation (17) gives us

$$\sum_c \sum_{ij} b_{ij}(\Phi_{cij}^* - \Phi_{cij}^p) \leq \frac{2C}{\lambda}\|\mathbf{b}\|^2.$$

We bound $\|\mathbf{b}\|^2$ with probability at least $1-\delta$ as given by Lemma 4.

$$\sum_c \sum_{ij} b_{ij}(\Phi_{cij}^* - \Phi_{cij}^p) \leq \frac{8(d+1)^4 C}{\epsilon^2 \lambda} \log^2\left(\frac{d}{\delta}\right). \tag{19}$$

Substituting this in Equation (14) proves the theorem.

□

The upper bound on the regularized empirical risk is in $O(\frac{C}{\epsilon^2})$. The bound increases for smaller values of ϵ which implies tighter privacy and therefore suggests a trade off between privacy and utility.

The regularized empirical risk of a classifier is calculated over a given training dataset. In practice, we are more interested in how the classifier will perform on new test data which is assumed to be generated from the same source as the training data. The expected value of the loss function computed over the data is called the *true risk* $\tilde{L}(\mathbf{\Phi}) = \mathbb{E}[L(\mathbf{\Phi})]$ of the classifier $\mathbf{\Phi}$. In the following theorem, we establish a bound on the true excess risk of the differentially private classifier minimizing the perturbed objective function and the classifier minimizing the original objective function.

Theorem 3. *With probability at least $1-\delta$, the true excess risk of the classifier $\mathbf{\Phi}^p$ minimizing the perturbed objective function $J_p(\mathbf{\Phi})$ over the classifier $\mathbf{\Phi}^*$ minimizing the unperturbed objective function $J(\mathbf{\Phi})$ is bounded as*

$$\tilde{L}(\mathbf{\Phi}^p) \leq \tilde{L}(\mathbf{\Phi}^*) + \frac{4\sqrt{d}(d+1)^2 C}{\epsilon\lambda}\log\left(\frac{d}{\delta}\right)$$
$$+ \frac{8(d+1)^4 C}{\epsilon^2\lambda}\log^2\left(\frac{d}{\delta}\right) + \frac{16}{\lambda n}\left[32 + \log\left(\frac{1}{\delta}\right)\right].$$

Proof. Let the expected value of the regularized empirical risk be

$$\tilde{J}(\mathbf{\Phi}) = \tilde{L}(\mathbf{\Phi}) + \lambda \sum_c \text{trace}(\mathbf{I}_{\mathbf{\Phi}}\mathbf{\Phi}_c\mathbf{I}_{\mathbf{\Phi}}). \tag{20}$$

Let $\mathbf{\Phi}^r$ be the classifier minimizing $\tilde{J}(\mathbf{\Phi})$, *i.e.*, $\tilde{J}(\mathbf{\Phi}^r) \leq \tilde{J}(\mathbf{\Phi}^*)$.
Rearranging the terms, we have

$$\begin{aligned}\tilde{J}(\mathbf{\Phi}^p) &= \tilde{J}(\mathbf{\Phi}^*) + [\tilde{J}(\mathbf{\Phi}^p) - \tilde{J}(\mathbf{\Phi}^r)] + [\tilde{J}(\mathbf{\Phi}^r) - \tilde{J}(\mathbf{\Phi}^*)] \\ &\leq \tilde{J}(\mathbf{\Phi}^*) + [\tilde{J}(\mathbf{\Phi}^p) - \tilde{J}(\mathbf{\Phi}^r)].\end{aligned}$$

Substituting the definition of $\tilde{J}(\mathbf{\Phi})$,

$$\tilde{L}(\mathbf{\Phi}^p) + \lambda \sum_c \text{trace}(\mathbf{I}_{\mathbf{\Phi}} \mathbf{\Phi}_c^p \mathbf{I}_{\mathbf{\Phi}}) \leq \tilde{L}(\mathbf{\Phi}^*) + \lambda \sum_c \text{trace}(\mathbf{I}_{\mathbf{\Phi}} \mathbf{\Phi}_c^* \mathbf{I}_{\mathbf{\Phi}}) + [\tilde{J}(\mathbf{\Phi}^p) - \tilde{J}(\mathbf{\Phi}^r)]$$

$$\tilde{L}(\mathbf{\Phi}^p) \leq \tilde{L}(\mathbf{\Phi}^*) + \lambda \sum_c \text{trace}[\mathbf{I}_{\mathbf{\Phi}} (\mathbf{\Phi}_c^* - \mathbf{\Phi}_c^p) \mathbf{I}_{\mathbf{\Phi}}] + [\tilde{J}(\mathbf{\Phi}^p) - \tilde{J}(\mathbf{\Phi}^r)]. \tag{21}$$

From Lemma 3 and Equation (18), we have,

$$\begin{aligned}\left[\sum_c \text{trace}[\mathbf{I}_{\mathbf{\Phi}} (\mathbf{\Phi}_c^* - \mathbf{\Phi}_c^p) \mathbf{I}_{\mathbf{\Phi}}]\right]^2 &\leq dC \sum_c \|\mathbf{\Phi}_c - \mathbf{\Phi}'_c\|^2 \\ &\leq \frac{4dC^2}{\lambda^2} \|\mathbf{b}\|^2 = \frac{16d(d+1)^4 C^2}{\epsilon^2 \lambda^2} \log^2\left(\frac{d}{\delta}\right).\end{aligned}$$

Taking the square root,

$$\sum_c \text{trace}[\mathbf{I}_{\mathbf{\Phi}} (\mathbf{\Phi}_c^* - \mathbf{\Phi}_c^p) \mathbf{I}_{\mathbf{\Phi}}] \leq \frac{4\sqrt{d}(d+1)^2 C}{\epsilon \lambda} \log\left(\frac{d}{\delta}\right). \tag{22}$$

Sridharan, *et al.* [17] present a bound on the true excess risk of any classifier as an expression of the bound on the regularized empirical excess risk for that classifier. With probability at least $1 - \delta$,

$$\tilde{J}(\mathbf{\Phi}^p) - \tilde{J}(\mathbf{\Phi}^r) \leq 2[J(\mathbf{\Phi}^p) - J(\mathbf{\Phi}^*)] + \frac{16}{\lambda n}\left[32 + \log\left(\frac{1}{\delta}\right)\right].$$

Substituting the bound from Theorem 2,

$$\tilde{J}(\mathbf{\Phi}^p) - \tilde{J}(\mathbf{\Phi}^r) \leq \frac{8(d+1)^4 C}{\epsilon^2 \lambda} \log^2\left(\frac{d}{\delta}\right) + \frac{16}{\lambda n}\left[32 + \log\left(\frac{1}{\delta}\right)\right]. \tag{23}$$

Substituting the results from Equations (22) and (23) into Equation (21) proves the theorem.

□

Similar to the bound on the regularized empirical excess risk, the bound on the true excess risk is also inversely proportional to ϵ reflecting the privacy-utility trade-off. The bound is linear in the number of classes C, which is a consequence of the multi-class classification. The classifier learned using a higher value of the regularization parameter λ will have a higher covariance for each class ellipsoid. This would also make the classifier less sensitive to the perturbation. This intuition is confirmed by the fact that the true excess risk bound is inversely proportional to λ.

6 Conclusion

In this paper, we present a discriminatively trained Gaussian classification algorithm that satisfies differential privacy. Our proposed technique involves adding a perturbation term to the objective function. We prove that the proposed algorithm satisfies differential privacy and establish a bound on the excess risk of the classifier learned by the algorithm which is inversely proportional to the data dimensionality which is directly proportional to the number of classes and inversely proportional to the privacy parameter ϵ reflecting a trade-off between privacy and utility.

In the future, we plan to extend this work along two main directions: extending our perturbation technique for a general class of learning algorithms and applying results from theory of large margin classifiers to arrive at tighter excess risk bounds for the differentially private large margin classifiers. Our intuition is that compared to other classification algorithms, a large margin classifier should be much more robust to perturbation. This would also give us insights into designing low error inducing mechanisms for differentially private classifiers.

Acknowledgements

We would like to thank the anonymous reviewers for their insightful comments.

References

1. Dwork, C.: Differential privacy. In: International Colloquium on Automata, Languages and Programming. (2006)
2. Chaudhuri, K., Monteleoni, C.: Privacy-preserving logistic regression. In: Neural Information Processing Systems. (2008) 289–296
3. McLachlan, G., Peel, D.: Finite Mixture Models. Wiley series in probability and statistics. Wiley-Interscience (2000)
4. Sha, F., Saul, L.K.: Large margin gaussian mixture modeling for phonetic classification and recognition. In: IEEE International Conference on Acoustics, Speech and Signal Processing. (2006) 265–268
5. Kasiviswanathan, S.P., Lee, H.K., Nissim, K., Raskhodnikova, S., Smith, A.: What can we learn privately? In: IEEE Symposium on Foundations of Computer Science. (2008) 531–540
6. Dinur, I., Nissim, K.: Revealing information while preserving privacy. In: Symposium on Principles of Database Systems. (2003)
7. Dwork, C., Nissim, K.: Privacy-preserving datamining on vertically partitioned databases. In: CRYPTO. (2004)
8. Blum, A., Dwork, C., McSherry, F., Nissim, K.: Practical privacy: The suLQ framework. In: Symposium on Principles of Database Systems. (2005)
9. Barak, B., Chaudhuri, K., Dwork, C., Kale, S., McSherry, F., Talwar, K.: Privacy, accuracy, and consistency too: a holistic solution to contingency table release. In: Symposium on Principles of Database Systems. (2007) 273–282

10. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Theory of Cryptography Conference. Volume 3876. (2006) 265–284
11. Jagannathan, G., Pillaipakkamnatt, K., Wright, R.N.: A practical differentially private random decision tree classifier. In: ICDM Workshop on Privacy Aspects of Data Mining. (2009) 114–121
12. Sha, F., Saul, L.K.: Large margin hidden markov models for automatic speech recognition. In: Neural Information Processing Systems. (2007) 1249–1256
13. Mahalanobis, P.C.: On the generalised distance in statistics. Proceedings of the National Institute of Sciences of India **2** (1936) 49–55
14. Chapelle, O.: Training a support vector machine in the primal. Neural Computation **19**(5) (2007) 1155–1178
15. Vandenberghe, L., Boyd, S.: Semidefinite programming. SIAM Review **38** (1996) 49–95
16. Chaudhuri, K., Monteleoni, C., Sarwate, A.D.: Differentially private empirical risk minimization. arXiv:0912.0071v4 [cs.LG] (2010)
17. Sridharan, K., Shalev-Shwartz, S., Srebro, N.: Fast rates for regularized objectives. In: Neural Information Processing Systems. (2008) 1545–1552

Appendix

Lemma 1. *Assuming all the data instances to lie within a unit ℓ_2 ball, the difference in the derivative of Huber loss function $L(\mathbf{\Phi}, \boldsymbol{x}, y)$ calculated over two data instances $(\boldsymbol{x}_i, y_i)$ and $(\boldsymbol{x}'_i, y'_i)$ is bounded.*

$$\left\| \frac{\partial}{\partial \mathbf{\Phi}_c} L(\mathbf{\Phi}, \boldsymbol{x}_i, y_i) - \frac{\partial}{\partial \mathbf{\Phi}_c} L(\mathbf{\Phi}, \boldsymbol{x}'_i, y'_i) \right\| \ \leq \ 2.$$

Proof. The derivative of the Huber loss function for the data instance $\boldsymbol{x}_i$ with label y_i is

$$\frac{\partial}{\partial \mathbf{\Phi}_c} L(\mathbf{\Phi}, \boldsymbol{x}_i, y_i) = \begin{cases} 0 & if\ \boldsymbol{x}_i^T(\mathbf{\Phi}_c - \mathbf{\Phi}_{y_i})\boldsymbol{x}_i > h, \\ \frac{1}{2h}[h - \boldsymbol{x}_i^T(\mathbf{\Phi}_{y_i} - \mathbf{\Phi}_c)\boldsymbol{x}_i]\boldsymbol{x}_i\boldsymbol{x}_i^T & if\ |\boldsymbol{x}_i^T(\mathbf{\Phi}_c - \mathbf{\Phi}_{y_i})\boldsymbol{x}_i| \leq h, \\ \boldsymbol{x}_i\boldsymbol{x}_i^T & if\ \boldsymbol{x}_i^T(\mathbf{\Phi}_c - \mathbf{\Phi}_{y_i})\boldsymbol{x}_i < -h. \end{cases}$$

The data points lie in a ℓ_2 ball of radius 1, $\forall i : \|\boldsymbol{x}_i\|_2 \leq 1$. Using linear algebra, it is easy to show that the Frobenius norm of the matrix $\boldsymbol{x}_i\boldsymbol{x}_i^T$ is same as the ℓ_2 norm of the vector $\boldsymbol{x}_i$, $\|\boldsymbol{x}_i\boldsymbol{x}_i^T\| = \|\boldsymbol{x}_i\|_2 \leq 1$.

As the term $\frac{1}{2h}[h - \boldsymbol{x}_i^T(\mathbf{\Phi}_{y_i} - \mathbf{\Phi}_c)\boldsymbol{x}_i]$ is at most one when $|\boldsymbol{x}_i^T(\mathbf{\Phi}_c - \mathbf{\Phi}_{y_i})\boldsymbol{x}_i| \leq h$, the Frobenius norm of the derivative of the Huber loss function is at most one in all cases, $\left\| \frac{\partial}{\partial \mathbf{\Phi}_c} L(\mathbf{\Phi}, \boldsymbol{x}_i, y_i) \right\| \leq 1$. Using a similar argument for data instance $\boldsymbol{x}'_i$ with label y'_i, we have $\left\| \frac{\partial}{\partial \mathbf{\Phi}_c} L(\mathbf{\Phi}, \boldsymbol{x}'_i, y'_i) \right\| \leq 1$.

Finally, using the triangle inequality $\|\boldsymbol{a} - \boldsymbol{b}\| = \|\boldsymbol{a} + (-\boldsymbol{b})\| \leq \|\boldsymbol{a}\| + \|\boldsymbol{b}\|$,

$$\begin{aligned} &\left\| \frac{\partial}{\partial \mathbf{\Phi}_c} L(\mathbf{\Phi}, \boldsymbol{x}_i, y_i) - \frac{\partial}{\partial \mathbf{\Phi}_c} L(\mathbf{\Phi}, \boldsymbol{x}'_i, y'_i) \right\| \\ &\leq \left\| \frac{\partial}{\partial \mathbf{\Phi}_c} L(\mathbf{\Phi}, \boldsymbol{x}_i, y_i) \right\| + \left\| \frac{\partial}{\partial \mathbf{\Phi}_c} L(\mathbf{\Phi}, \boldsymbol{x}'_i, y'_i) \right\| \leq 2. \end{aligned}$$

□

Lemma 2. *The objective function $J(\mathbf{\Phi})$ is λ-strongly convex. For $0 \leq \alpha \leq 1$,*

$$J\left(\alpha\mathbf{\Phi} + (1-\alpha)\mathbf{\Phi}'\right) \leq \alpha J(\mathbf{\Phi}) + (1-\alpha)J(\mathbf{\Phi}') - \frac{\lambda\alpha(1-\alpha)}{2}\sum_c \left\|\mathbf{\Phi}_c - \mathbf{\Phi}'_c\right\|^2.$$

Proof. By definition, Huber loss is λ-strongly convex, *i.e.*

$$L\left(\alpha\mathbf{\Phi} + (1-\alpha)\mathbf{\Phi}'\right) \leq \alpha L(\mathbf{\Phi}) + (1-\alpha)L(\mathbf{\Phi}') - \frac{\lambda\alpha(1-\alpha)}{2}\left\|\mathbf{\Phi} - \mathbf{\Phi}'\right\|^2. \quad (24)$$

where the Frobenius norm of the matrix set $\mathbf{\Phi} - \mathbf{\Phi}'$ is the sum of norms of the component matrices $\mathbf{\Phi}_c - \mathbf{\Phi}'_c$,

$$\left\|\mathbf{\Phi} - \mathbf{\Phi}'\right\|^2 = \sum_c \left\|\mathbf{\Phi}_c - \mathbf{\Phi}'_c\right\|^2. \quad (25)$$

As the regularization term $N(\mathbf{\Phi})$ is linear,

$$\begin{aligned} N(\alpha\mathbf{\Phi} + (1-\alpha)\mathbf{\Phi}') &= \lambda\sum_c \mathrm{trace}(\alpha\mathbf{I}_{\mathbf{\Phi}}\mathbf{\Phi}_c\mathbf{I}_{\mathbf{\Phi}} + (1-\alpha)\mathbf{I}_{\mathbf{\Phi}}\mathbf{\Phi}'_c\mathbf{I}_{\mathbf{\Phi}}) \quad (26)\\ &= \alpha\lambda\sum_c \mathrm{trace}(\mathbf{I}_{\mathbf{\Phi}}\mathbf{\Phi}_c\mathbf{I}_{\mathbf{\Phi}}) + (1-\alpha)\lambda\sum_c \mathrm{trace}(\mathbf{I}_{\mathbf{\Phi}}\mathbf{\Phi}'_c\mathbf{I}_{\mathbf{\Phi}})\\ &= \alpha N(\mathbf{\Phi}) + (1-\alpha)N(\mathbf{\Phi}'). \end{aligned}$$

The lemma follows directly from the definition $J(\mathbf{\Phi}) = L(\mathbf{\Phi}) + N(\mathbf{\Phi})$.

□

Lemma 3.

$$\frac{1}{dC}\left[\sum_c \mathit{trace}[\mathbf{I}_{\mathbf{\Phi}}(\mathbf{\Phi}_c - \mathbf{\Phi}'_c)\mathbf{I}_{\mathbf{\Phi}}]\right]^2 \leq \sum_c \left\|\mathbf{\Phi}_c - \mathbf{\Phi}'_c\right\|^2$$

Proof. Let $\Phi_{c,i,j}$ be the $(i,j)^{\text{th}}$ element of the size $(d+1)\times(d+1)$ matrix $\mathbf{\Phi}_c - \mathbf{\Phi}'_c$. By the definition of the Frobenius norm, and using the identity $N\sum_{i=1}^N x_i^2 \geq (\sum_{i=1}^N x_i)^2$,

$$\begin{aligned} \sum_c \left\|\mathbf{\Phi}_c - \mathbf{\Phi}'_c\right\|^2 &= \sum_c \sum_{i=1}^{d+1}\sum_{j=1}^{d+1} \Phi^2_{c,i,j} \geq \sum_c \sum_{i=1}^{d+1} \Phi^2_{c,i,i} \geq \sum_c \sum_{i=1}^{d} \Phi^2_{c,i,i}\\ &\geq \frac{1}{dC}\left(\sum_c \sum_{i=1}^{d} \Phi_{c,i,i}\right)^2 = \frac{1}{dC}\left[\sum_c \mathrm{trace}[\mathbf{I}_{\mathbf{\Phi}}(\mathbf{\Phi}_c - \mathbf{\Phi}'_c)\mathbf{I}_{\mathbf{\Phi}}]\right]^2. \end{aligned}$$

□

Lemma 4.

$$P\left[\|\mathbf{b}\| \geq \frac{2(d+1)^2}{\epsilon}\log\left(\frac{d}{\delta}\right)\right] \leq \delta.$$

Proof. Similar to the union bound argument used in Lemma 5 in [2].